Information Leakage Prevention in IPsec VPN

Vishnu K G[1], Chinchu Jacob[2]

[1] Student of KMP College of Engineering, Perumbavoor, Ernakulam

[2] Assistant Professor at KMP College of Engineering, Perumbavoor, Ernakulam.

Abstract: ***Virtual Private Networks are most widely used in all over the world. According to VPN it is considered that the networks are secure. But in all cases there are some loop holes to penetrate into it. So in this paper we discussing about the most dangerous loop holes namely the covert channels. There are a lot of research is done in the field of covert channels. But all works are not practically implemented to get an idea of how to prevent the covert channel. In this we are specifying the threat model and using some algorithms to prevent the covert channels. By the use of this concept we can completely prevent the information leakage (almost 90%).***

Keywords: IPsec, VPN, Covert channel, Information Leakage

1. Introduction

Virtual Private Networks (VPNs) are used to secure the information passed through the networks. Using VPN we can reduce the leakage of information from the network. In the basic VPN systems the local area networks are connected through VPN gateways. The VPN gateways are always secure. Also in this basic idea we using the IPsec security. So that the information transferred are more secure.

Virtual private networks are famous to safely join their system destinations over the Internet. Their security is executed and authorized by VPN Gateway that passes the information through secure channels.

Covert channels abuse the framework security approach by utilizing channels "not expected for data exchange ". While there is a wide variety of exploration on covert channels, few works have considered the useful execution and execution effect of far reaching in modern networks in present day systems. We believe this problem is important for various reasons, particularly in virtual systems and VPNs

Insider Threat: As opposed to end-to-end secure channels, where the endpoints are verifiably trusted, VPNs are additionally utilized for consistent system detachment and edge security requirement. In this setting, the individuals from a VPN are frequently not completely trusted, but rather the trust is reduced to authorization focuses, the VPN gateways, which should avoid undesired data streams. On the other hand, insiders in the LAN may leak data through the VPN gateway utilizing secret channels, in this way bypassing the security strategy. Illustrations of such insiders can be genuine people or stealth malware, participating in mechanical secret activities, leaking real time financial exchange information, or disclosing a lot of information from physically secured industry.

Traffic Analysis: By investigating traffic examples and metadata, it is additionally conceivable to derive data about exchanged information without expecting a malicious insider. Such "detached" Man-in-the-Middle (MITM) situations are turning out to be more predominant with network virtualization, permitting co-located, as far as anyone knows separated frameworks to examine one another.

Combination with Detection: Although application-layer firewalls and IDS frameworks are generally conveyed, carefully implemented covert channels stay hard to distinguish. In these frameworks, the insider picks a weaker flag and mirrors the patterns of regular channel utilization.

Covert channel prevention can be valuable here to induce noise, forcing the enemy to utilize a more grounded sign and in this way encourage discovery. We expect the combination of covert channel mitigation and discovery to take into consideration less intrusive example implementation and in this way fundamentally reduce the execution penalty.

Outline: After stating about the present threat model in the VPN the next section will discuss about the covert channel identification methods. In the next section we will discuss about the covert channel prevention mechanism and also insider threat detection method.

2. Adversary Model

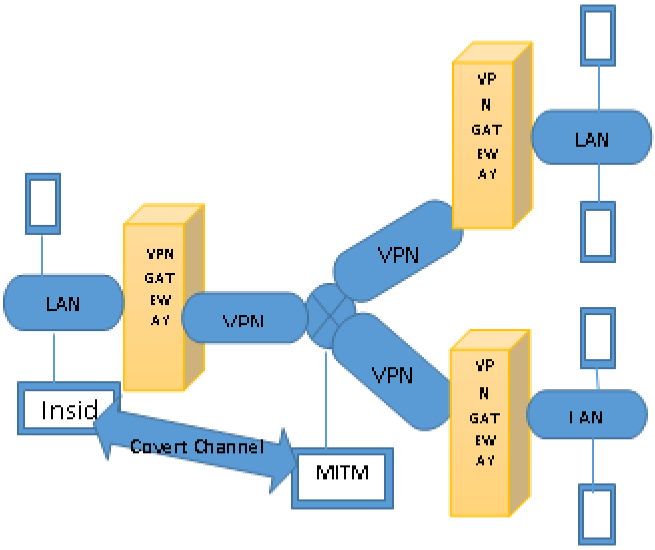


Figure 1: Adversary model

Considering a threat model which consist of some insiders and man in the middle. Insiders are treated as a software or an event taken place in a particular time or any person. So we can say insider as a compromised host. And man in the middle, one who is active outside the local area network. The MITM monitor the traffic in the entire VPN network. The goal of the attacker is to create an unauthenticated communicating channel between the compromised host and the man in the middle.

The model of an adversary created channel (Covert channel) is illustrated in the above figure Fig.1. By the use of this channel the adversaries can communicate each other and transfer the information. So an outside attacker can easily get the information from the network. Information are transferred through the unauthenticated channel. The covert channels are hard to detect. So we assume that the messages are encrypted through IPsec and the encapsulated security payload. Also the VPN gateways are enforced to check the cryptographic primitives.

The attacker controls one or more compromised host has in the LAN locales and additionally an active MITM in the WAN. We refer to the LAN hosts controlled by the attacker as insiders, paying little heed to whether they are controlled by genuine people or malware. The attacker's objective is to build up a correspondence channel between the MITM and one or all the possible malicious insiders, as outlined in Fig. 1. This would permit the attacker to send guidelines to the insiders or to break data from the secured to the unprotected area, breaking the edge security of the VPN. For this reason, we expect a best in class IPsec design with confirmed encryption utilizing Encapsulated Security Payload (ESP) in tunnel mode [17], and the cryptographic primitives of the VPN are safely upheld by the VPN gateway. Then again, the legitimate VPN activity can be controlled by attackers in the ensured and unprotected spaces to trade data that "survives" these packet change upheld by the VPN gateways.

Unfortunately, no orderly approach is known for distinguishing system covert channels separated from thorough searching channels, and the arrangement as capacity or timing channels can be uncertain [5].We utilized a far reaching investigation on the IPsec detail and related deal with covert channels directs in network protocols, and also source code examination and testing to distinguish potential covert channels in IPsec VPNs. IP-Tunnelling and confirmed encryption by the IPsec gateways incredibly rearranged this issue, as none of the protocol headers that the MITM can read or change (i.e., the external IP and ESP header) are straightforwardly accessible to the LAN hosts.

Altogether, we have distinguished just eight secret channels. The accessible covert channels involve three storage based channels situated in light of fields in the external IP header (ECN, DS, Flags) and five timing-based covert channels that control Inter-Packet Delay (IPD), packet order (PktOrd), WAN limit (PktDrop), and Path MTU Discovery (PMTUD). Staying normal for the individual destination LAN of a packet (DestIP) does not constitute an covert channel in its own privilege however can go about as enhancement of other covert channels.

3. Covert Channel Detection

In this area we exhibit the outline of identifying covert channels using strong IPsec, i.e., a framework with low, known covert channel capacity and high throughput. We present novel or enhanced strategies for efficient covert channel detection. We determine all the inbound and outbound covert channel capacity of the network. Also limiting the covert channel capacity to almost zero.

Packet Size (PktSize): The packet size trademark is normally tended to by padding packets to most extreme size or expecting them to be of consistent size [6]. Be that as it may, as the item throughput = pkt_size·pkt_rate is consistent for a given connection, requirement of small packet sizes can lessen the load per packet fundamentally, permitting higher packet rates and more synchronous associations

It was already proposed to permit numerous other packet sizes, however then the proportion between packets of diverse sizes makes another covert channel. Mode Security was proposed to deal with the exchanging between diverse requirement modes and review such a staying clandestine channel. In any case, genuine system movement is regularly monitored, i.e., packet streams utilizing distinctive packet sizes are frequently transmitted in the meantime.

In addition, the requirement of little packet sizes is risky for IP conventions: With Path MTU Discovery (PMTUD), the network endpoints rapidly recognize and adjust to the greatest permitted packet size of an IP course, yet just gradually recover to a bigger MTU utilizing a moderate trial and error approach. This dynamic adaption likewise makes it harder for the VPN gateways to appraise the real interest for packets of bigger size.

We address these issues by consolidating packet padding with straightforward discontinuity and multiplexing, instruments that were beforehand considered for traffic pattern obfuscation. Packet parts inside IPsec permits us to proficiently and straightforwardly enforce different packet sizes at the gateway without impacting the channel's Path MTU (PMTU). This is not quite the same as standard IP parts before or after IPsec preparing, which brings about noticeable sections either on the LAN or WAN sides that could again be utilized as covert channels. The fragments instrument is supplemented by packet multiplexing, which can be utilized to less packet padding overhead by linking numerous small packets up to the coveted packet size. This likewise lessens the IPsec encapsulation overhead (ESP, IP).

At the point when working with mixed traffic, the sender gateway first pieces expansive packets and after that endeavours to multiplex little packets or parts into the padding territory of already prepared packets that are still in the packet support. At the receiving gateway, packets are first de-multiplexed and afterward defragmented. As this component work straightforwardly for the LAN sender and collector, the LAN gateways can accurately screen the present requests of the contiguous LAN site to ideally modify the enforced packet size.

Inter Packet Delay (IPD): The clandestine channel in view of IPDs and its alleviation were subject of a few past works (e.g., [6], [10], [12]–[14]). In principle, it is effortlessly dispensed with by enforcing a fixed IPD at the VPN gateway, embedding dummy packets when no genuine packets are accessible [24]. In any case, because of the high packet rates in advanced systems, even brief times of non-ideal IPDs (and hence packet rate) requirement at the VPN gateway rapidly bring about packet loss because of buffer overflows or system clogging. This is frequently disturbed by the sent clogging shirking calculations, which will rapidly adjust to a lower enforced packet rate while just gradually making utilization of expanded rates, along these lines making it troublesome for the VPN gateway to assess the ideal authorization rate.

The impact can be somewhat relieved with bigger packet supports; on the other hand, this can likewise make high packet delays, debasing system responsiveness [15]. Further, the ideal enforced packet rate can be expansive in modern systems, making a high computational overhead for the time synchronous packet preparing. Case in point, to soak a 100 Mbit/s join with 200 byte packets, a normal IPD of 500 byte/100·106 byte/s = 2µs ought to be enforced. At long last, one must consider mistakes in the timing requirement that show up at high framework loads [3], [2]:

Since high movement on the LAN interface can impact the framework heap of the passage, a LAN host may actuate errors in the IPD implementation of the gateway that can again be measured by the MITM, yielding CIPD r = 0.16 bps

We have actualized traffic reshaping inside the Linux bit, utilizing the bit's High-Precision Event Timer (HPET) base for packet scheduling with nanosecond determination. This generously decreases the overhead of setting exchanging and buffering, permitting IPDs in the scope of microseconds instead of a few milliseconds (e.g., [9], [12]) and detectably enhances throughput and responsiveness.

To keep up great framework execution at significantly higher packet rates we utilize packet bursts, i.e., we make an interpretation of low IPDs into bursts of various packets at correspondingly bigger delays. For ideal packet buffering we alter the buffer size contingent upon the right authorized IPD. This anticipates long postpones at low rates while permitting liberal buffering at high rates.

Packet Order (PktOrd): Sequence numbers in protocol headers have been utilized before to make an covert or stenographic channel in light of packet reordering [6], [7]. Be that as it may, rather than past works we can take out this divert in the VPN situation utilizing the IPsec anti- replay window and secure sequence numbers in ESP.

IPsec usage as of now keep up a bitmap of the last r seen and inconspicuous sequence numbers so replay attacks inside of the window size can be identified. To dispose of communication through packet re-requesting, we propose to actualize this window as a packet buffer, where new packets are embedded sorted by their ESP sequence number. At the point when a validated packet with a higher sequence number than the presently viewed as set of r is watched, the window advances to that most elevated seen number and any packets dropping out of the support are sent. Accordingly, all packets sent from the VPN gateway into the LAN are requested, paying little respect to packet drops, and the covert channel is disposed of: CPktOrd r,in = 0.

But, the methodology is risky for low packet rates, following the window may progress gradually and singular packets are not sent sufficiently and quickly. We tackle this by setting up a certain greatest IPD (e.g., 50ms, which likewise guarantees system responsiveness at low throughput modes), and by having a gateway sent at any rate r dummy packets before a network can be stopped.

Packet Drops (PktDrop): as a rule, it seems difficult to dispense with covert channels in view of packet dropping in the WAN. Alleviation with lapse error codes is costly and effectively defeated by dropping significantly more packets. Rather, we propose to relieve the channel by infusing noise, by expanding packet loss relatively to the genuine packet loss. In particular, let the gateways keep up a buffer D of size l. At the sender gateway, packets are buffered in D and their request is randomized before exemplification.

At the beneficiary gateway, the packets are again gathered in D and the quantity of dropped packets i in a sequence of l packets is resolved taking into account their ESP sequence number. In the event that i > 0, the gateway drops another j packets from the present buffer, such that i + j = 2x, for 1 < x ≤ log2(l), and advances the remaining packets in the way of randomizing their request once again. Thus, the MITM can pick the general number of packets to be dropped for the getting LAN customer however can't choose which packets to drop, bringing about an image space of log2(l) + 1 packets for every l packets. The remaining covert channel limit is then CPktDrop r,in = 1l · log2(log2(l) + 1) b

Like in the packet re-requesting relief, the inbound packet buffer D at the accepting passage is dangerous for low packet rates and requires comparable confinements to guarantee a constant flow of packets. Note that usage could consolidate the inbound packet sorting and IPD authorization with the packet dropping facility to relieve the delays of utilizing various queues.

Path MTU Discovery (PMTUD): as far as anyone is concerned, no past work thought about how possible it is of covert channels in view of PMTUD, specifically as for VPNs. Since PMTUD is discriminating for good system execution, we don't impair it however rather relieve the channel by implementing points of confinement on the rate and qualities that are spread by the VPN gateways into the LAN.

Specifically, we limit the conceivable PMTU values by keeping up a rundown of regular PMTU qualities and just propagate the individual next lower PMTU to the LAN. Such normal PMTUs qualities can be set up on location or can be gotten from previously proposed execution improvements for PMTUD [8]. The rate limitation of PMTU spread is problamistic when all is said in done, as an absence of MTU adaption will prompt packet loss. On the other hand, for our situation the current PMTU is constantly known not trusted VPN gateways, which can then utilize the straightforward fracture highlight from PktSize authorization to interpret in the middle of LAN and WAN packet sizes. Considering the 10 most regular PMTU qualities and a normal interim of, e.g., 2 minutes [8] between spread of PMTU changes, our measures lessen the covert channel rate to not exactly CPMTUD r,in = 0.03 bps.

Storage Based Channels (ECN, DS, Flags): The storage based covert channels misusing the Explicit Congestion Notification (ECN), Differentiated Services (DS) and IPv4 Flags treatment of IP/IPsec are effectively dispensed with by resetting the particular fields of the external IP header at exemplification and overlooking them amid de-capsulation. Normalizing the IPv4 Flags field is unproblematic as on the way fragments is censured in IP. Be that as it may, disposing of the ECN and DS covert channels impairs these execution enhancements in the WAN.

Active Probing: The MITM may deduce data on the LAN status by effectively examining the IPsec gateways and assessing their reaction conduct, a methodology that was presented as dynamic activity investigation [2], [4]. Specifically, a LAN customer could bring about high load on the LAN interface of the IPsec passage. The subsequent change in the passage's framework burden can then be measured by how it reacts to authentic administration asks for by the MITM, for example, ICMP pings [3]. Note that, in spite of the past channel attributes, this assault really abuses a side channel at the passage: Its ability does not rely on upon the utilization of the VPN channel however on the recurrence at which the insider can affect high and low framework loads at the gateway and on the rate at which the MITM has the storage test the gateway to gauge its framework load with adequate exactness.

4. Covert Channel Prevention

Covert channel prevention is a main concern in the case of IPsec VPN to reduce the information leakage through the network. Basically there are two types of covert channels, inbound covert channel and the outbound covert channel. Inbound covert channel is the channel used to pass the information from the man in the middle to the insider. Outbound covert channel is the channel used to pass the information from insider to the man in the middle. Insider detection and preventing the covert channel is the main concern. Outbound covert channel is the most information leakage area because of the information from the insider. So more care is taken to this channel.

- Pseudo code of Local Flow Monitoring Algorithm

```
set Xval 0
 for {set i 1} {$i<=$cv} {incr i} {
 set X($i) [$null($i) set npkts_]
 puts "X($i):$X($i)"
 set Xval [expr $Xval+$X($i)]
}

puts "Xval:$Xval"
for {set i 1} {$i<=$cv} {incr i} {
set P($i) [expr ($X($i)+0.0)/$Xval]
puts "P($i):$P($i)"
}
proc log {base x} {
set lv [expr {log($x)/log($base)}]
return $lv
}

set H 0
for {set i 1} {$i<=$cv} {incr i } {
if {$P($i)!=0} {
set log_val [log 2 $P($i)]
set H [expr $H+($P($i)*($log_val+0.0))]
}
}

puts "Hval:$H"
incr iteration
set c 0
for {set i 0} {$i<5} {incr i} {
set null($i) [new Agent/LossMonitor]
}
```

We are using the real time methods because of effectiveness. For the covert channel prevention using the local flow monitoring algorithm. The algorithm is implemented with all the security methods to prevent the information leakage. In the algorithm creating a dummy queue with a fixed size and place it in the network. So that the attacker only get the information from the dummy queue. Also implementing the security policies and security rules to monitor the information get in to the network and goes out. Each information packets are fragmented and adding the security parameter. Security parameter will identify the changes to the packet such as the packet size, packet order, inter packet delay etc.

5. Detection and Prevention of Insider

Insiders are most dangerous in case of a secured network. In the case of VPN, insider is the only way to leaking information to the outsiders. So as the VPN we have to identify the insiders in the system, which can be identified by using the IP trace back algorithm. In the algorithm it checks for all malicious activities across the VPN network from the abnormal behavior of the hosts. It may leads to the information leakage.

- Pseudo code of IP Trace back Algorithm

```
set A [list]
set diff_val [expr $H-$c_val]
set flag_detected 0
$ns at $time "$ns trace-annotate \"The Entropy value is:$H\""
$ns at $time "$ns trace-annotate \"The del value is $del_val\""
if {$diff_val>$del_val} {
 $ns at $time "$ns trace-annotate \"Attack detected during interval [expr ($delT*$iteration)-$delT]- [expr $delT*$iteration]\""
 set flag_detected 1
 }

 if {$flag_detected==1} {
 for {set i 1} {$i<=6} {incr i} {
 foreach att $attacker {
 $ns at [$ns now] "$n($att) color red"
 }
 }
 }

 $ns at [expr $delT*$iteration] "PacketRateAdjustment"
}
$ns at $delT "PacketRateAdjustment"
proc Quantize {arnew arq} {
 global rmax rnew Mr
 set rq [expr $rmax/($Mr-1)]
 set rnow [expr $arnew/($arq+1)]
}
```

Insider should be considered as the harmful parties of the network. So they should be eliminated from the network. Whenever the threat or the insider party should be identified it should be noticed to the system administrator. The system administrator should remove all the privileges of the host and remove from the entire network. So the insiders should loss all the access to the network.

Insiders are identified by the behavior of the host. Whenever an information is transferring from one network to another network the VPN gateway broadcasts the information to the receiving host's local network. So the corresponding host will access the packet and others will reject the packet without any changes to the packet. While the insiders accept the packet, packets should be transferred to the external attacker. Attacker can sometimes crack the message and reveal the actual data in the encrypted message. Insiders are identified and removed from the network by using the criteria such as the monitoring the entire network, algorithmic procedures. Insider prevention should be taken place by using enforced security procedures and implementing the security policies and principals.

6. Performance Evaluation

6.1 Testbed and Raw Performance

In this area we describe the execution accomplished by our model regarding network throughput, transaction rate (i.e., roundtrip time) and protocol overhead. Our testbed compares to the VPN situation in Fig. 1, aside from that we utilize just two LAN destinations with one physical host for each LAN. The Man-in-the-Middle (MITM) is actualized as an Ethernet bridge between the two VPN portals, permitting dependable perception of every single transmitted packet. For our assessment, the MITM is totally uninvolved and just used to give free execution estimations of the WAN. All hosts are 3.2 Ghz Intel Core i5-650 machines, outfitted with two Intel PCIe GBit system cards and 4GB framework memory. All system connections are built up at full-duplex GBit/s speed.

We list the general execution results in Table III. The initial two sections demonstrate the testbed execution for raw IP (plaintext) transmission and IPsec ESP tunneling. With 570 Mbit/s, the raw transmission does not achieve the normal GBit throughput, likely because of lacking equipment or drivers. As the LAN hosts and the MITM measure the same IP payloads, there is no LAN/WAN overhead. With 201 Mbit/s, the throughput of a standard IPsec ESP gateway is as of now prominently slower because of 10% protocol overhead yet chiefly computational imperatives of the VPN gateways. As our secretive channel relief is an augmentation of this ESP tunnel arrangement, we standardize the relative throughput to 100%.

For reference and confirmation of the normal execution overhead of our model, we next assessed the raw execution of our HPCM Engine contrasted with the standard IPsec ESP tunnel. The third segment "TFC" of Table III records the accomplished system execution when tunneling TFC inside ESP with all covert channel alleviation methods handicapped. The general LAN/WAN overhead of 13% (or 3% when contrasted and the ESP gateway) is the consequence of the 8 to 12 byte TFC protocol epitome in addition to some computational overhead.

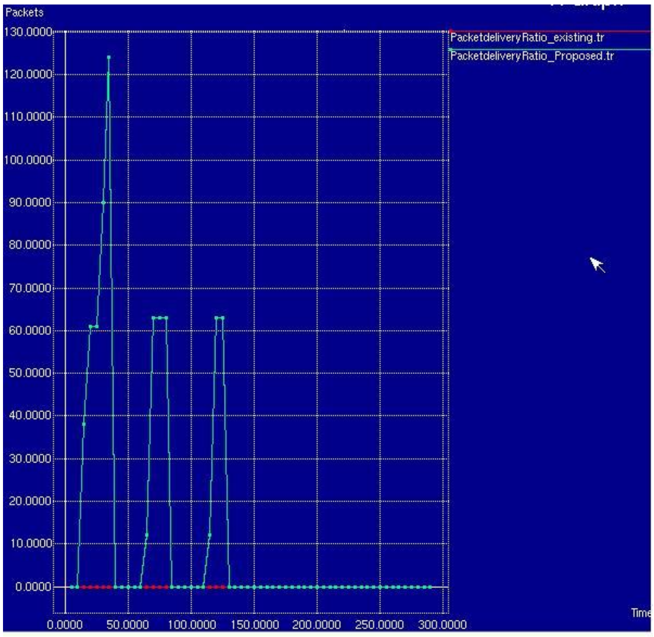
6.2 Experimental Results

We will assess our proposed scheme through both hypothetical examination and simulations. Assessing the information leakage prevention through the existing and proposed IPsec VPN. The examination is in light of practically identical security levels. In the experiment we calculate the packet delivery ratio, throughput, packet loss ratio and end to end delay.

6.2.1 Packet Delivery Ratio

The degree of the quantity of conveyed information packet to the destination. This represents the level of conveyed information to the destination. Packet conveyance degree is more critical on account of the message verification and transmission from a sender to the recipient

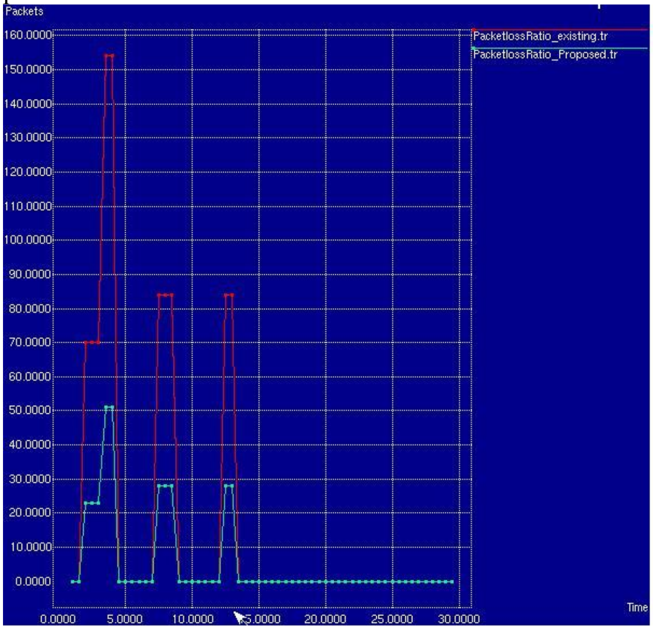
Packet Delivery Ratio = Total number of packet receive / Total number of packet send



6.2.2 Packet Loss Ratio

The aggregate number of packets dropped amid the reproduction. At the point when packets are sends to a destination from a source the packets may be misfortunes by the activity of assailants. This is likewise computed in the packet misfortune proportion in our simulation technique.

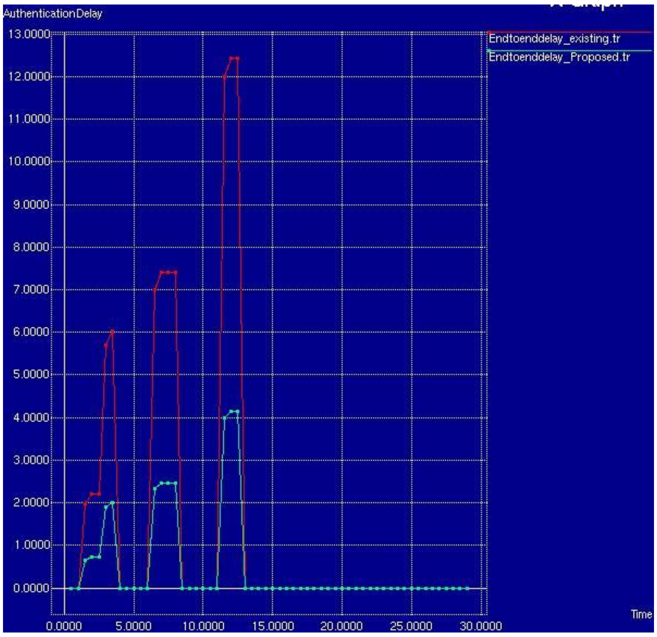
Packet lost = Total number of packet send – Total number of packet received.



6.2.3 End to end delay

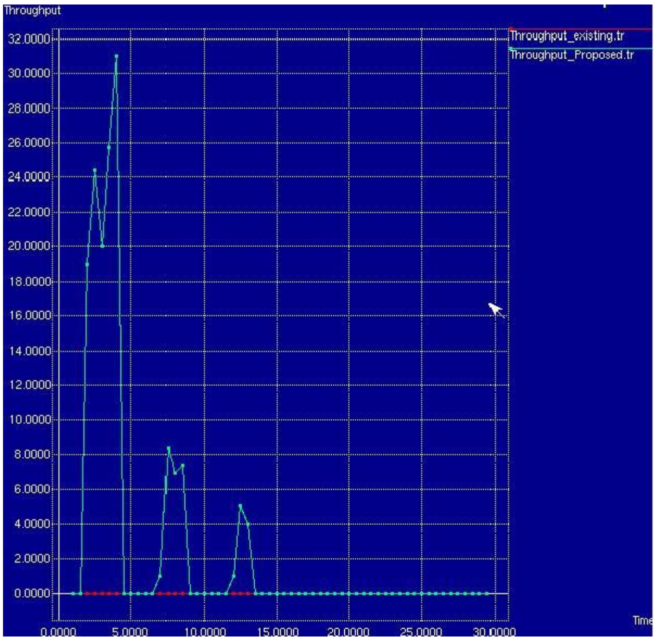
The normal time taken by an information packet to touch base in the destination. It likewise incorporates the postponement created by course revelation methodology and the line in information packet transmission. Just the information packets that effectively conveyed to destinations that tallied.

End to End delay = Sum (arrive time – send time) / ∑Total number of connections



6.2.4 Throughput

It is the number of data packets successfully send through the network communication channel.



7. Conclusion

We have defined the problem of covert channels in the IPsec VPN. Presented a novel architecture of our adversary model and determining how it will happening in our real time network. Presenting the techniques to identify the covert channels, preventing the covert channels and detection and prevention of the insiders. After designing these we are performing a analysis for the correctness of our work.

References

[1] Cohesive Flexible Technologies, Chicago, IL, USA. (2012, Apr.). *VPN-Cubed* [Online]. Available: http://cohesiveft.com

[2] L. Catuogno, A. Dmitrienko, K. Eriksson, D. Kuhlmann, G. Ramunno, A.-R. Sadeghi, *et al.*, ―Trusted virtual domains—Design, implementation and lessons learned," in *Proc. Int. Conf. Trusted Syst.*, 2009, pp. 156–179.

[3] J. Carapinha, P. Feil, P. Weissmann, S. Thorsteinsson, Ç. Etemo˘glu, O. Ingthórsson, *et al.*, ―Network virtualization—Opportunities and challenges for operators," in *Proc. FIS*, 2010, pp. 138–147.

[4] B. R. Venkatraman and R. E. Newman-Wolfe, ―Capacity estimation and auditability of network covert channels," in *Proc. IEEE Symp. Security Privacy*, May 1995, pp. 186–198.

[5] M. Liberatore and B. N. Levine, ―Inferring the source of encrypted HTTP connections," in *Proc. Conf. CCS*, 2006, pp. 255–263.

[6] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, ―Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds," in *Proc. 16th ACM Conf. CCS*, 2009, pp. 199–212.

[7] B. Graham, Y. Zhu, X. Fu, and R. Bettati, ―Using covert channels to evaluate the effectiveness of flow confidentiality measures," in *Proc. 11th ICPADS*, Jul. 2005, pp. 57–63.

[8] Y. Liu, D. Ghosal, F. Armknecht, A.-R. Sadeghi, S. Schulz, and S. Katzenbeisser, ―Hide and seek in time—Robust covert timing channels," in *Proc. Eur. Symp. Res. Comput. Security*, 2009, pp. 120–135.

[9] S. Murdoch and S. Lewis, ―Embedding covert channels into TCP/IP," in *Information Hiding*. New York, NY, USA: Springer-Verlag, 2005.

[10] B. R. Venkatraman and R. E. Newman-Wolfe, ―Performance analysis of a method for high level prevention of traffic analysis using measurements from a campus network," in *Proc. 10th Annu. Comput. Security Appl. Conf.*, Dec. 1994, pp. 288–297.

[11] J. Millen, ―20 years of covert channel modeling and analysis," in *Proc. IEEE Symp. Security Privacy*, May 1999, pp. 113–114.

[12] C. Kiraly, S. Teofili, R. Lo Cigno, M. Nardelli, and E. Delzeri, ―Traffic flow confidentiality in IPsec: Protocol and implementation," in *The Future of Identity in the Information Society*. New York, NY, USA: Springer-Verlag, 2008.

[13] C. V. Wright, S. E. Coull, and F. Monrose, ―Traffic morphing: An efficient defense against statistical traffic analysis," in *Proc. NDSS*, 2009, pp. 1–14.

[14] S. Moskowitz and A. R. Miller, ―Simple timing channels," in *Proc. IEEE Comput. Soc. Symp. Res. Security Privacy*, May 1994, pp. 56–64.

[15] X. Fu, ―On traffic analysis attacks and countermeasures," Ph.D. dissertation, Texas A&M University, College Station, TX, USA, Dec. 2005.\

[16] B. R. Venkatraman and R. E. Newman-Wolfe, ―Transmission schedules to prevent traffic analysis," in *Proc. 9th ACSAC*, 1994, pp. 108–115.

[17] J. Mogul and S. Deering, ―Path MTU discovery," RFC Rep. 1191, Nov. 1990.

Author Profile

Vishnu K G received the B.Tech degree in Computer Science and Engineering from Mahathma Gandhi University, kottayam, Kerala, India in 2012 and currently pursuing final year M. Tech degree in Computer Science and Engineering with specialization in Cyber Security from KMP College of Engineering, Perumbavoor. Also his research areas are network security, information/data security and cryptography.

Chinchu Jacob received B.Tech in Computer Science and Engineering and M.Tech in Computer Science and Engineering with specialization in Data Security from Cochin University of Science and Technology, Cochin, Kerala, India in 2010 and 2012 and currently working as assistant professor in KMP College of Engineering Perumbavoor in Computer Science and Engineering Department.